<u>IN THE CLAIMS</u>

1. (Original) A method for processing client requests supporting a plurality of object models, the method comprising:

receiving a former client request requiring access to a former object defined by a former object model;

mapping a former object required for access by the former client request to a corresponding current object existing within a current object model;

copying current object data within the current object of the current object model to former object data within an instantiation of the former object; and

processing the former client request using the instantiation of the former object to satisfy the former client request.

2. (Original) The method of claim 1 wherein receiving a former client request requiring access to a former object defined by a former object model comprises:

exposing a former service interface for use by former clients for receipt of former client requests;

receiving the former client request on the former service interface; and

forwarding the former client request via the former service interface to an former client adapter for processing of the former client request.

3. (Original) The method of claim 2 wherein exposing a former service interface for use by former clients for receipt of former client requests comprises:

providing a former remote method invocation interface for former clients to use to provide former client requests for processing; and

concurrently providing a current remote method invocation interface for current clients to use to provide current client requests for processing.

4. (Original) The method of claim 2 wherein mapping a former object required for access by the former client request to a corresponding current object existing within a current object model comprises:

obtaining an identity of the former object specified within the former client request;

obtaining a former object definition from the identity of the former object within a former object model definition; and

using the former object definition, instantiating the former object within a memory system.

5. (Original) The method of claim 4 wherein:

the former object and current object are defined in an object-oriented programming language; and

wherein obtaining a former object definition comprises:

using a reflection technique to identify, from a former object model definition file, the former object definition based on the identity of the former object specified within the former client request; and

wherein instantiating the former object within a memory system comprises:

using a former object class loader to load the former object definition, identified in the former object definition file, into the memory system as an instantiation of the former object.

6. (Original) The method of claim 4 wherein mapping a former object required for access by the former client request to a corresponding current object existing within a current object model comprises:

accessing object mapping entries in an object map to identify a current object definition defined in a current object model that corresponds to the instantiated former object required for access by the former client request; and

instantiating the current object based on the current object definition.

7. (Original) The method of claim 6 wherein copying current object data within the current object of the current object model to former object data within an instantiation of the former object comprises:

copying current object data values stored within current data fields of an instantiation of the current object to former data values within former data fields of an instantiation of the former object.

8. (Original) The method of claim 7 wherein copying current object data values stored within current data fields of an instantiation of the current object to former data fields within an instantiation of the former object comprises:

identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, and

i) if an object adapter exists, operating the adapter to convert the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object; and

ii) if no object adapter exists, copying the current field data in the instantiation of the current object for use as former field data for use in the instantiation of the former object.

9. (Original) The method of claim 8 wherein identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object comprises:

using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name;

and if the field adapter class exists, loading and operating a method in the field adapter class to perform the conversion of the current field data in the

instantiation of the current object to former field data for use in the instantiation of the former object

10. (Original) The method of claim 9 wherein using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name comprises:

obtaining the predetermined name for use in the reflect technique from at least one of:

a name of the former object;

a name of a field of data within the former object;

a name of the current object;

a name of a field of data within the current object; and

a field name identified in object mapping entries in an object map used for mapping the current object to the former object.

11. (Original) The method of claim 1 wherein copying current object data within the current object of the current object model to former object data within an instantiation of the former object comprises:

copying current object data values stored within current data fields of an instantiation of the current object to former data values within former data fields of an instantiation of the former object.

12. (Original) The method of claim 11 wherein copying current object data values stored within current data fields of an instantiation of the current object to former data fields within an instantiation of the former object comprises:

identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, and

i) if an object adapter exists, operating the adapter to convert the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object; and

ii) if no object adapter exists, copying the current field data in the instantiation of the current object for use as former field data for use in the instantiation of the former object.

13. (Original) The method of claim 12 wherein identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object comprises:

using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name; and

if the field adapter class exists, loading and operating a method in the field adapter class to perform the conversion of the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object

14. (Original) The method of claim 1 wherein:

the operations of receiving a former client request, mapping a former object to a corresponding current object, copying current object data within the current object to former object data within an instantiation of the former object and processing the former client request are performed within a current version of a storage area network management application; and

wherein receiving a former client request comprises:

receiving the former client request from a former version of a storage area network management client application, the former client request containing a request for access to former storage area network management object data

contained within former objects defined by a former object model that has been replaced by the current object model; and

wherein the steps of mapping, copying and processing the former client request allow the former version of the storage area network management client application to operate with the current version of the storage area network management application that maintains storage area network management data within current objects conforming to the current object model.

15. (Original) A computer system comprising:

a memory;

a processor;

a communications interface;

an interconnection mechanism coupling the memory, the processor and the communications interface;

wherein the memory is encoded with a server application including an object adapter that when performed on the processor, is operable to process client requests supporting a plurality of object models by causing the computer system to perform the operations of:

receiving, via the communications interface into the memory, a former client request requiring access to a former object defined by a former object model;

mapping a former object required for access by the former client request to a corresponding current object existing within a current object model;

copying current object data within the current object of the current object model to former object data within an instantiation of the former object; and

processing the former client request using the instantiation of the former object to satisfy the former client request.

16. (Original) The computer system of claim 15 wherein when the service application causes the computer system to perform the operation of receiving a

former client request requiring access to a former object defined by a former object model, the service application causes the computer system to perform the operations of:

exposing a former service interface for use by former clients for receipt of former client requests;

receiving the former client request on the former service interface; and

forwarding the former client request via the former service interface to an former client adapter for processing of the former client request.

17. (Original) The computer system of claim 16 wherein when the service application causes the computer system to perform the operation of exposing a former service interface for use by former clients for receipt of former client requests, the service application causes the computer system to perform the operations of:

providing a former remote method invocation interface for former clients to use to provide former client requests for processing; and

concurrently providing a current remote method invocation interface for current clients to use to provide current client requests for processing.

18. (Original) The computer system of claim 16 wherein when the service application causes the computer system to perform the operation of mapping a former object required for access by the former client request to a corresponding current object existing within a current object model, the service application causes the computer system to perform the operations of:

obtaining an identity of the former object specified within the former client request;

obtaining a former object definition from the identity of the former object within a former object model definition; and

using the former object definition, instantiating the former object within a memory system.

19. (Original) The computer system of claim 18 wherein:

the former object and current object are defined in an object-oriented programming language; and

wherein when the service application causes the computer system to perform the operation of obtaining a former object definition, the service application causes the computer system to perform the operation of:

using a reflection technique to identify, from a former object model definition file, the former object definition based on the identity of the former object specified within the former client request; and

wherein when the service application causes the computer system to perform the operation of instantiating the former object within a memory system, the service application causes the computer system to perform the operation of:

using a former object class loader to load the former object definition, identified in the former object definition file, into the memory system as an instantiation of the former object.

20. (Original) The computer system of claim 18 wherein when the service application causes the computer system to perform the operation of mapping a former object required for access by the former client request to a corresponding current object existing within a current object model the service application causes the computer system to perform the operation of:

accessing object mapping entries in an object map to identify a current object definition defined in a current object model that corresponds to the instantiated former object required for access by the former client request; and

instantiating the current object based on the current object definition.

21. (Original) The computer system of claim 20 wherein when the service application causes the computer system to perform the operation of copying current object data within the current object of the current object model to former

object data within an instantiation of the former object, the service application causes the computer system to perform the operation of:

copying current object data values stored within current data fields of an instantiation of the current object to former data values within former data fields of an instantiation of the former object.

22. (Original) The computer system of claim 21 wherein when the service application causes the computer system to perform the operation of copying current object data values stored within current data fields of an instantiation of the current object to former data fields within an instantiation of the former object, the service application causes the computer system to perform the operations of:

identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, and

i) if an object adapter exists, operating the adapter to convert the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object; and

ii) if no object adapter exists, copying the current field data in the instantiation of the current object for use as former field data for use in the instantiation of the former object.

23. (Original) The computer system of claim 22 wherein when the service application causes the computer system to perform the operation of identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, the service application causes the computer system to perform the operations of:

using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name; and

if the field adapter class exists, loading and operating a method in the field adapter class to perform the conversion of the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object

24. (Original) The computer system of claim 23 wherein when the service application causes the computer system to perform the operation of using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name, the service application causes the computer system to perform the operation of:

obtaining the predetermined name for use in the reflect technique from at least one of:

a name of the former object;

a name of a field of data within the former object;

a name of the current object;

a name of a field of data within the current object; and

a field name identified in object mapping entries in an object map used for mapping the current object to the former object.

25. (Original) The computer system of claim 15 wherein when the service application causes the computer system to perform the operation of copying current object data within the current object of the current object model to former object data within an instantiation of the former object, the service application causes the computer system to perform the operation of:

copying current object data values stored within current data fields of an instantiation of the current object to former data values within former data fields of an instantiation of the former object.

26. (Original) The computer system of claim 25 wherein when the service application causes the computer system to perform the operation of copying current object data values stored within current data fields of an instantiation of the current object to former data fields within an instantiation of the former object, the service application causes the computer system to perform the operation of:

identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, and

i) if an object adapter exists, operating the adapter to convert the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object; and

ii) if no object adapter exists, copying the current field data in the instantiation of the current object for use as former field data for use in the instantiation of the former object.

27. (Original) The computer system of claim 26 wherein when the service application causes the computer system to perform the operation of identifying, for each current field in the current object that maps to a corresponding former field in the former object, if a field adapter exists for converting current field data in the instantiation of the current object to former field data for use in the instantiation of the former object, the service application causes the computer system to perform the operations of:

using a reflection technique to identify, within a set of adapter classes, if a field adapter class exists that corresponds to a predetermined name; and

if the field adapter class exists, loading and operating a method in the field adapter class to perform the conversion of the current field data in the instantiation of the current object to former field data for use in the instantiation of the former object

28. (Original) The computer system of claim 15 wherein:

the operations of receiving a former client request, mapping a former object to a corresponding current object, copying current object data within the current object to former object data within an instantiation of the former object and processing the former client request are performed within a current version of a storage area network management application executing on the computer system; and

wherein when the service application causes the computer system to perform the operation of receiving a former client request, the service application causes the computer system to perform the operation of:

receiving the former client request from a former version of a storage area network management client application, the former client request containing a request for access to former storage area network management object data contained within former objects defined by a former object model that has been replaced by the current object model; and

wherein the steps of mapping, copying and processing the former client request allow the former version of the storage area network management client application, operating on a computer system in communication with the computer system operating the server application, to operate with the current version of the storage area network management application that maintains storage area network management data within current objects conforming to the current object model.

29. (Canceled)

30. (Original) A computer program product having a computer-readable medium including computer program logic encoded thereon that, when executed on a computer system, processes client requests supporting a plurality of object models by causing the computer system to perform the operations of:

receiving a former client request requiring access to a former object defined by a former object model;

mapping a former object required for access by the former client request to a corresponding current object existing within a current object model;

copying current object data within the current object of the current object model to former object data within an instantiation of the former object; and

processing the former client request using the instantiation of the former object to satisfy the former client request.

31. (Previously Presented) A method for processing client requests supporting a plurality of object models, the method comprising:

receiving a plurality of requests from former client versions requiring access to respective former objects defined by respective former object models, wherein the object models are shared object models;

exposing former service interfaces for use by former client versions for receipt of requests from former client versions;

mapping former objects, required for access by the requests from former client versions, to a corresponding current object existing within a current object model, wherein the current object model is a shared model, and wherein the current object model is not directly backwards compatible with the requests from former client versions;

wherein mapping former objects further comprises indicating current objects that correspond to former objects defined in a former object definition file, and data within the current objects that correspond to data in the former objects;

copying current object data from fields of the current object of the current object model to former object data within an instantiation of the former objects; and

processing the requests from former client versions using the instantiation of the former objects to satisfy the former client requests, thereby providing backwards compatibility.